Compressed Sensing and Dictionary Learning to Alleviate Tradeoff between Temporal and Spatial Resolution in Videos

Pranav Kulkarni, 15D070017
Anmol Kagrecha, 15D070024
Karan Taneja, 15D070022

I. Introduction

Digital cameras face a fundamental trade-off between spatial and temporal resolution. This limitation is primarily due to hardware factors such as the readout and time for analog-to-digital (A/D) conversion in image sensors. The problem can be alleviated by using parallel A/D converters and frame buffers but it incurs more cost.

Hitomi et al. [1] and Liu et al. [2] propose techniques for sampling, representing and reconstructing the space-time volume in order to overcome this trade-off. We aim to propose and experiment with different sampling strategies which can improve the performance but are not necessarily constrained by hardware implementations.

The techniques used by Hitomi et al. [1] and Liu et al. [2] contain several hyper-parameters. We also aim to understand the effect of these hyper-parameters on the performance of the techniques by performing exhaustive experiments.

II. Overview of the Approach

Denote the space-time volume corresponding to an $M \times M$ pixel neighborhood and one frame integration time of the camera as $E(x, y, t)$. A projection of this volume along the time dimension is captured by a camera. A N times gain in temporal resolution needs to be achieved, i.e., recovery of space-time volume E needs to be done at a resolution of $M \times M \times N$. Let $S(x, y, t)$ denote the per-pixel shutter function of the camera within the integration time ($S(x, y, t) \in \{0, 1\}$). Then, the captured image $I(x, y)$ is

$$I(x, y) = \sum_{t=1}^{N} S(s, y, t).E(x, y, t). \tag{1}$$

Rewriting the (1) in the matrix form as $\mathbf{I} = \mathbf{SE}$. $\mathbf{I}$ which is the vector of observations has a size M^2 and $\mathbf{E}$ which is the vector of unknowns has a size $N \times M^2$. To solve the under-determined system of equations above, techniques of compressed sensing are used. The sytem above can be solved faithfully if the signal $\mathbf{E}$ has a sparse representation $\boldsymbol{\alpha}$ in a dictionary $\mathbf{D}$:

$$\mathbf{E} = \mathbf{D}\boldsymbol{\alpha} = \alpha_1\mathbf{D}_1 + \cdots + \alpha_k\mathbf{D}_k \tag{2}$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_k]^T$ are the coefficients, and $\mathbf{D}_1, \cdots, \mathbf{D}_k$ are the elements in dictionary $\mathbf{D}$. The coefficient vector $\boldsymbol{\alpha}$ is sparse. The over-complete dictionary $\mathbf{D}$ is learned from a random collection of videos. The space-time volume $\mathbf{E}$ is sampled with a coded exposure function $\mathbf{S}$ and then projected along the time dimension, resulting in a coded exposure image $\mathbf{I}$.

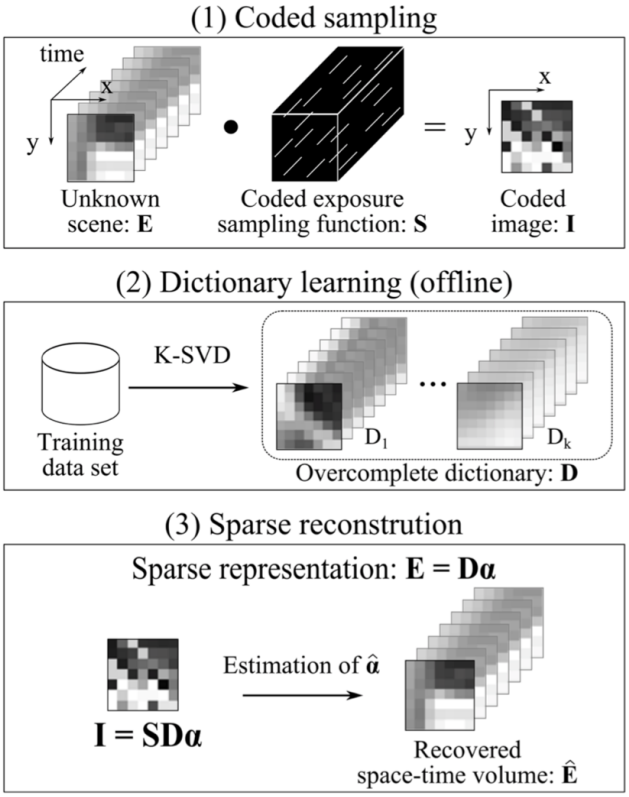

Fig. 1: Overview of the approach [1]

An estimate of the coefficient vector $\hat{\boldsymbol{\alpha}}$ can be obtained by solving a standard problem in compressed sensing:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\alpha} \|\alpha\|_0 \text{ subject to } \|\mathbf{SD}\boldsymbol{\alpha} - \mathbf{I}\|_2^2 < \varepsilon \tag{3}$$

The space-time volume is computed as $\hat{\mathbf{E}} = \mathbf{D}\hat{\boldsymbol{\alpha}}$. A pictorial representation of the approach containing three steps viz. coded sampling, dictionary learning and sparse reconstruction is given in figure 1.

III. Sampling Strategies

A. Continuous Bump

The continuous bump sampling strategy is a strategy that satisfies hardware constraints and is proposed by Liu et al. It satisfies the following constraints:

- **Binary shutter**: The sampling function **S** takes values in $\{0, 1\}$. At any time t, a pixel is either collecting light (1-on) or not (0-off).
- **Single bump exposure**: Each pixel can have only one continuous on time (i.e., a single bump) during a camera integration time.
- **Fixed bump length**: The bump length is fixed for all pixels.

We implement this strategy as follows:

1) For each pixel in the input video, the bump start time is selected uniformly randomly, i.e., the bump start time at any pixel can be t with probability $1/N$ where N is the temporal depth of **E**.
2) Space time volume at a pixel is added when the shutter function is 1. The coded image is constructed by averaging the above sum.

B. Random Sampling

In this sampling strategy, the bump period may not be continuous. Fix a bump length b and let the temporal depth be N, then this strategy is implemented as follows:

1) For every point in an input video, i.e., a point described by (x, y, t), the shutter function takes value 1 with probability b/N.
2) Space time volume at a pixel is added when the shutter function is 1. The coded image is constructed by averaging the above sum.

It might happen for a pixel location that the number of time instants where shutter function takes value 1 is not equal to the bump length.

C. Distributed Bump

Here, we constrain that for every pixel location, the number of time instants where shutter function takes value 1 is equal to the bump length. This is implemented as follows:

1) Generate b distinct random integers between 1 and N.
2) Space time volume at a pixel is added when the shutter function is 1. The coded image is constructed by averaging the above sum.

IV. Details of Algorithms Used

A. Orthogonal Matching Pursuit (OMP)

We use the OMP[3] algorithm for both dictionary learning as well as for reconstruction, i.e, finding the coefficient vector $\boldsymbol{\alpha}$. The algorithm is given in Algorithm 1.

Here **A** is the measurement matrix, **y** is the measurement vector and T_0 is the sparsity constraint. For the purpose of reconstruction, the measurement matrix **A** is **DS** and the measurement vector **y** is **I**.

Algorithm 1 OMP algorithm

procedure OMP($\mathbf{A}, \mathbf{y}, T_0$)
$S^0 \leftarrow \emptyset$
$\mathbf{x}^0 \leftarrow \mathbf{0}$
for $i \in \{1, ..., T_0\}$ **do**
$j_{n+1} \leftarrow \arg\max_j \left\{|(\mathbf{A}^*(\mathbf{y} - \mathbf{A}\mathbf{x}^i))_j|\right\}$
$S^{n+1} \leftarrow S^n \cup \{j_{n+1}\}$
$\mathbf{x}^{n+1} \leftarrow \arg\min_{\mathbf{z}\in\mathbb{C}^n} \left\{\|\mathbf{y}-\mathbf{A}\mathbf{z}\|_2, \text{supp}(\mathbf{z}) \subset S^{n+1}\right\}$

B. K-SVD

K-SVD [4] is used to learn the dictionary from the training videos. The algorithm is given in Algorithm 2. The task is to find the best dictionary to represent the data samples $\{\mathbf{y}_i\}_{i=1}^N$ as sparse compositions by solving

$$\min_{\mathbf{D},\mathbf{X}}\{||\mathbf{Y} - \mathbf{D}\mathbf{X}||_F^2\} \text{ subject to } \forall i, ||\mathbf{x}_i||_0 \leq T_0$$

Algorithm 2 K-SVD algorithm

procedure KSVD

Set the dictionary matrix $\mathbf{D}^{(0)} \in \mathbb{R}^{n\times k}$ with l^2 normalized columns

Set J=1

while stopping rule is not satisfied **do**

Use any pursuit algorithm to compute representation vectors $\mathbf{x}_i$, for each example $\mathbf{y}_i$, by approximating the solution of

$$i = 1, 2, \ldots, N, \min_{\mathbf{x}_i}\{||\mathbf{y}_i - \mathbf{D}\mathbf{x}_i||_2^2\} \text{ subject to } ||\mathbf{x}_i||_0 \leq T_0$$

For each column $k = 1, 2, \ldots, K$ in $D^{(J-1)}$, update it by:

— Define the group of examples that use this atom $\omega_k = \{i|1 \leq i \leq N, \mathbf{x}_T^k(i) \neq 0\}$

— Compute the overall representation error matrix by $\mathbf{E}_k = \mathbf{Y} - \sum_{j\neq k} \mathbf{d}_j \mathbf{x}_T^J$

— Restrict $\mathbf{E}_k$ by choosing only the columns corresponding the ω_k, and obtain $\mathbf{E}_k^R$

— Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\Delta\mathbf{V}^T$. Choose the update dictionary column d_k to be the first column of **U**. Update the coefficient vector $\mathbf{x}_R^k$ to be the first column of V multiplied by $\Delta(1, 1)$

Set J = J + 1.

V. Experiments Performed

A. Setup

Given is an overview of our experimental setup:

1) We collected a high temporal resolution (1000 fps) video containing a variety of scenes.
2) We sample 20 videos having the number of frames equal to the temporal depth. The starting times of these videos are selected randomly.

3) We fix the number of dictionary elements to be learned for each video. The data provided to K-SVD is a collection of patches which have been rotated in 8 directions.
4) Dictionary elements for each video are then learned using K-SVD and appended to get a final dictionary.
5) We then perform coded sampling on the training videos to obtain the training coded images. We also perform coded sampling on a set of test videos to obtain the test coded images.
6) Reconstruction is done patch-wise using the OMP algorithm. The overlap between the patches is controlled by the parameter 'stride'.
7) Finally we calculate the relative mean squared error (RMSE) between the original video $\mathbf{V}$ and the reconstructed video $\hat{\mathbf{V}}$. RMSE is defined as:

$$\text{RMSE} = \frac{\|\mathbf{V} - \hat{\mathbf{V}}\|_F^2}{\|\mathbf{V}\|_F^2} \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm.

B. Implementational details

The setup has 4 main components. Prepossessing of videos performed to normalize each video and reduce spatial dimension to size 160x320. A function to generate a coded image is implemented which takes in as input the pre-processed videos and parameters depending on the sampling scheme. Another function is implemented which can learn the dictionary using the KSVD algorithm based on the parameters provided such as sparsity level and the number of basis in the dictionary. A module to reconstruct video from the coded aperture uses OMP to find sparse representation in the learn dictionary basis. We use the libraries provided in [5] for KSVD and OMP, as these were much more efficient in their execution time that the functions written by us.

Following are the typical parameters that we worked with. In each of the subsequent experiments, we vary one or two of these parameters at a time to observe their effect on reconstruction error. Image size is 160x320 with temporal depth is set to 36. The sparsity of the videos is assumed to be 40, and 625 basis learned from each of the 20 videos and appended together to get the dictionary. We use patch size of 8 and stride of 4 in our experiments. Bump length for generating the coded image is taken to be 3.

C. Parameters

The parameters of the experiments performed are:

- Temporal depth
- Sparsity
- Bump length
- Number of basis per video segment
- Patch size
- Stride
- Noise variance in the coded image
- Sampling function

D. Noise Variance and Bump Length

We increase the bump length from 1 to 5 and the reconstruction gets better as the bump length increases. The sampling strategy here uses a continuous bump.

After a point increase in bump length (towards $S(x, y, t) = 1$) is expected to increase the RMSE. As time progresses, the pixel values are less likely to be correlated. If highly uncorrelated values are added, it is going to lead to a loss of information.

RMSE also increases with increase in noise variance. This is expected because more noise is will lead to a greater loss of information. The trends for the training and test data are shown in figure 2.

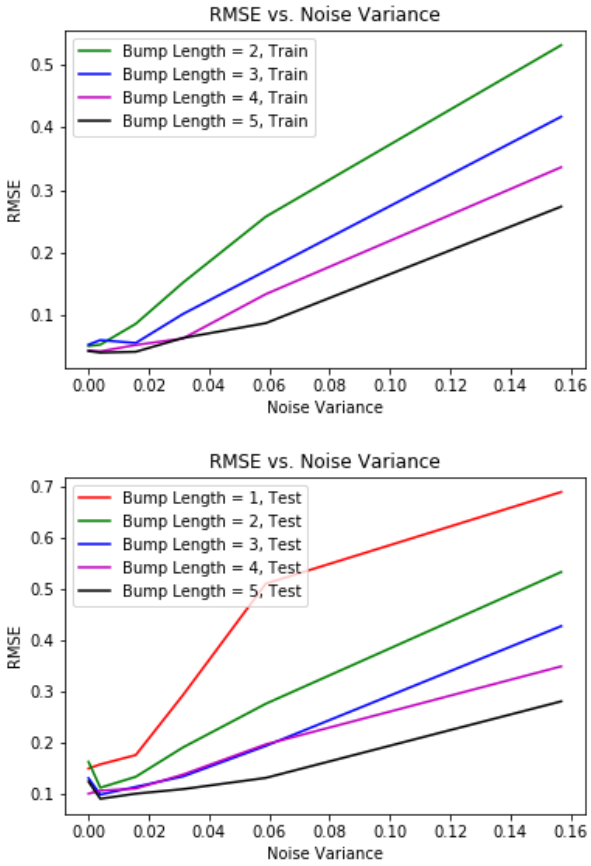

Fig. 2: Effect of Noise Variance and Bump Length

E. Temporal Depth

We fix the sampling strategy to continuous bump. We observe that the RMSE increases with an increase in temporal depth. This is because a greater number of elements need to be recovered from the same amount of evidence. The trends for the training and test data are shown in figure 3.

F. Sparsity and Number of Basis Elements in Dictionary

We trained two dictionaries having 325 and 625 number of basis elements per video. We fix the sampling strategy to have a continuous bump.

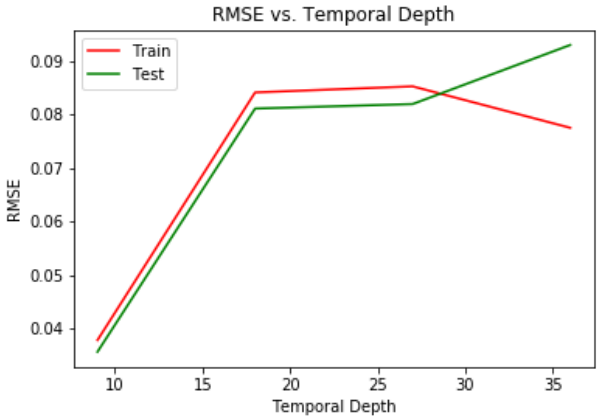

Fig. 3: Effect of Temporal Depth

Surprisingly, the smaller dictionary had a smaller RMSE but the difference is not very large. Further experimentation is required to ascertain the effects of the change in the number of basis elements in the dictionary.

Increase in sparsity decreased the RMSE. It is expected to saturate as the sparsity is increased further. The trends for the training and test data are shown in figure 4.

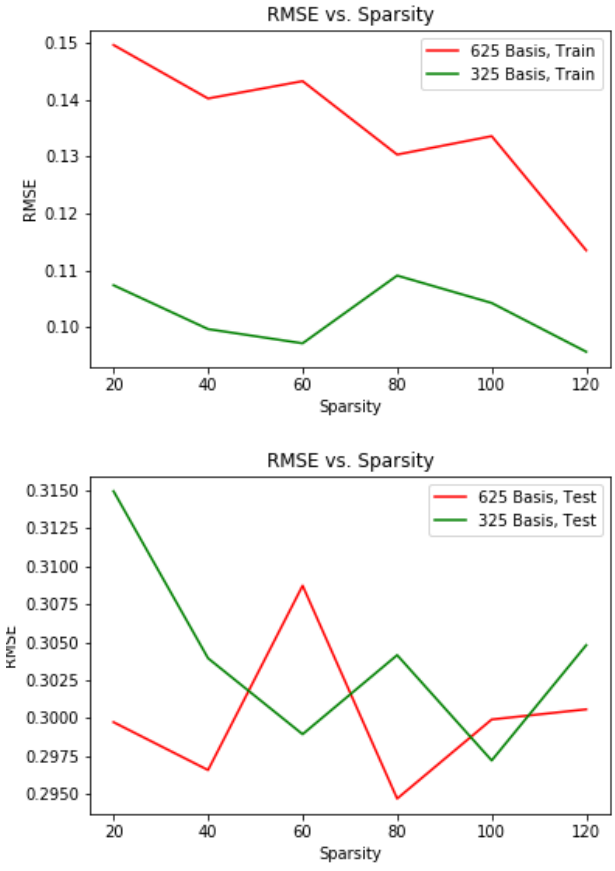

Fig. 4: Effect of Sparsity and number of Basis Elements

G. Patch Size and Stride

We fix the sampling strategy to have a continuous bump. We use a dictionary with 625 basis elements per video.

The decrease in stride decreases the RMSE because of more overlap between neighboring patches. The patchiness in the video reconstructed is also less.

Increasing patch size decreases the RMSE because each patch captures more information. However, this trend of RMSE with patch size is expected to saturate unless the number of basis in also increased. The results can be seen in figure 5.

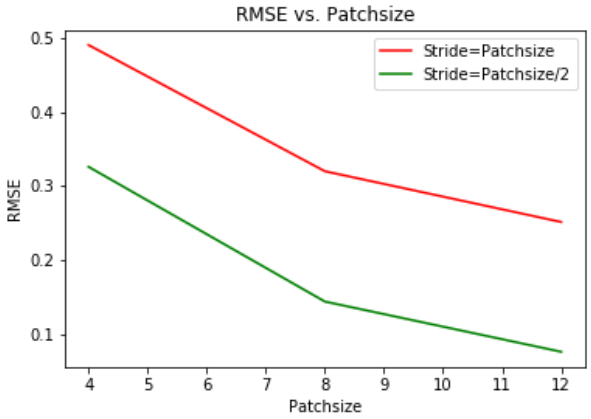

Fig. 5: Effect of Patch Size and Stride

H. Sampling Strategies

We finally compare the sampling strategies. We construct a dictionary with 625 basis elements per video. The patch size is 8×8 and the stride is 4.

We observe that the random sampling strategy performs the worst. This happens because the number of points where the information is getting collected may be much less than bump length times the number of pixels. Some of the spatial locations may not be sampled at all anywhere in their temporal depth, leading to complete loss of information.

Distributed bump strategy performs the best. It is a pseudo-random policy with an important constraint. For each pixel, the number of time instants where information is captured equals the bump length. The requirement for continuity of bump length is relaxed. The trends can be observed in figure 6.

VI. Conclusion

This method, that we have experimented with in this project, can reconstruct the videos with high temporal resolution without compromising on spatial resolution, though some artifacts are visible, most likely because of smaller dictionary size that we have used. We observed the effect of changing different hyper-parameters. We also observed that distributed bump sampling produces the best results, but this has to be at the cost of increased hardware complexity.

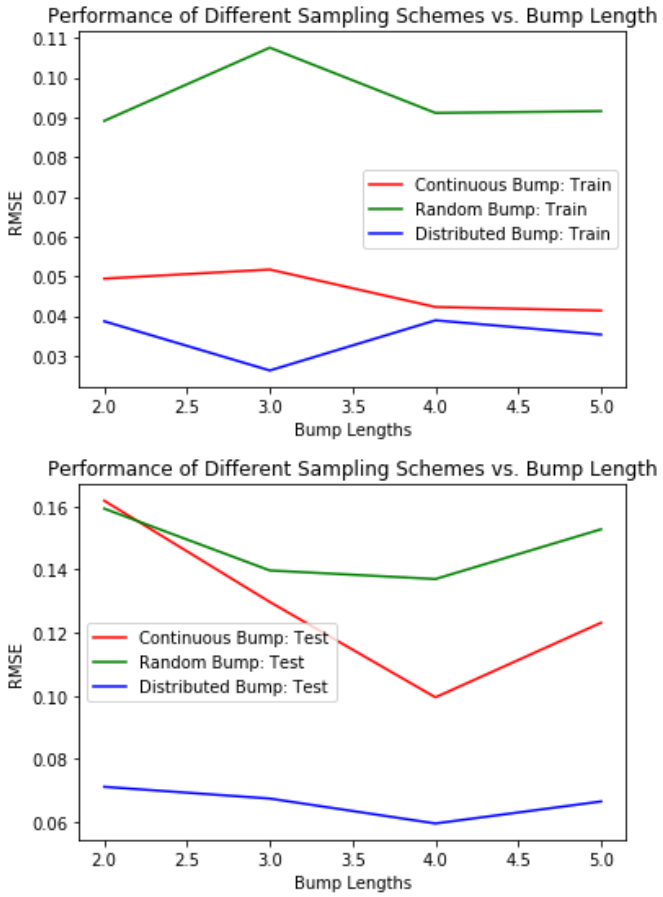

Fig. 6: Performance of different sampling strategies

REFERENCES

[1] Yasunobu Hitomi, Jinwei Gu, Mohit Gupta, Tomoo Mitsunaga, Shree K. Nayar (2011) *Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary*

[2] Dengyu Liu, Jinwei Gu, Yasunobu Hitomi, Mohit Gupta, Tomoo Mitsunaga, Shree K. Nayar (2014) *Efficient Space-Time Sampling with Pixel-wise Coded Exposure for High-Speed Imaging*

[3] Simon Foucart, Holger Rauhut (2013) *A Mathematical Introduction to Compressive Sensing*

[4] Michal Aharon, Michael Elad, and Alfred Bruckstein (2006) *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation*

[5] Implemntation of KSVD and OMP: *http://www.cs.technion.ac.il/ ronrubin/software.html*